CLAIMS

1. A method of synchronizing at least two concurrently running processes in a data processing system, comprising:

(a) providing a first array of elements with initialized states, each element of said first array having a concurrently running process associated therewith, each element of said first array being configured to have its state updated by its associated concurrently running process upon completion of a phase by said associated concurrently running process;

(b) providing a second array of elements with initialized hold states, each element of said second array having a concurrently running process associated therewith, each element of said second array being configured to switch, upon receiving an instruction, to a release state;

(c) arranging for monitoring said first array of elements and, upon each element of said first array having had its state updated, arranging for generating said instruction for switching said elements of said second array to said release state.

2. The method recited in claim 1, further comprising:

(d) for each process of said at least two concurrently running processes, configuring said each process such that, upon completion of said phase and upon updating of its associated element of said first array, said each process then waits at its associated element of said second array for said release state.

3. The method recited in claim 2, wherein each element of said first array has a byte size corresponding to the size of a cache line used in said data processing system.

4. The method recited in claim 3, wherein each element of said second array has a byte size corresponding to the size of said cache line used in said data processing system.

5. The method recited in claim 4, further comprising providing each element of said second array locally in relation to its respective, associated process.

6. The method recited in claim 2, further comprising, upon said each element of said first array having had its state updated, and prior to generating said instruction for switching said elements of said second array to said release state, arranging for reinitializing each element of said first array.

7. The method recited in claim 1, wherein in (c), said monitoring of said first array of elements is performed by one of said concurrently running processes.

8. The method recited in claim 1, wherein in (c), said monitoring of said first array of elements is performed by an independent process.

9. The method recited in claim 1, wherein in (a), said initialized state of said each element of said first array is a value.

10. The method recited in claim 1, wherein in (a), said each element of said first array comprises a state machine.

11. The method recited in claim 10, wherein said state machine is one of a counter, a gate, a flag and a sensor.

12. The method recited in claim 1, wherein in (b), said each element of said second array comprises a state machine.

13. The method recited in claim 12, wherein said state machine is one of a counter, a gate, a flag and a sensor.

14. A system for synchronizing at least two concurrently running processes in a data processing system, comprising:

(a) a first array of elements, each element of said first array having a concurrently running process associated therewith, said each element of said first array being configured to have an initial state that may be updated by its associated concurrently running process, upon completion of a phase by said associated concurrently running process;

(b) a second array of elements, each element of said second array having a concurrently running process associated therewith, said each element of said second array being configured to have an initial hold state that may be switched, upon receiving an instruction, to a release state;

(c) a monitoring process for monitoring said first array of elements, said monitoring process being configured to generate said instruction for switching said elements of said second array to said release state, upon each element of said first array having had its state updated.

15. The system recited in claim 14, wherein each element of said first array has a byte size corresponding to the size of a cache line used in said data processing system.

16. The system recited in claim 15, wherein each element of said second array has a byte size corresponding to the size of said cache line used in said data processing system.

17. The system recited in claim 14, wherein each element of said second array is provided locally in relation to its respective, associated process.

18. The system recited in claim 14, wherein said each element of said first array is a state machine.

19. The system recited in claim 14, wherein said each element in one of a counter, a gate, a flag and a switch.

20. The system recited in claim 14, wherein said each element of said second array is a state machine.

21. The system recited in claim 14, wherein said state machine is one of a counter, a gate, a flag, a switch, and a sensor.

22. The system recited in claim 14, wherein said at least two concurrently running processes execute on multiple processors embodied within a single computer.

23. The system recited in claim 14, wherein said at least two concurrently running processes execute on multiple processors distributed across multiple computers connect across a network.

24. A processor for executing a process in order to synchronize said process with at least one other concurrently running process, said processor being operable to:

access an element of a first array of elements, said element of said first array being associated with said process, said element of said first array having an initial state;

update said element of said first array of elements upon completion of a phase by said process;

after said updating, access an element of a second array of elements, said element of said second array being associated with said process, said element of said second array having an initial hold state and being configured to switch, upon receiving an instruction, to a release state, and check said element of said second array for said switch to said release state until detecting said release state.

25. A method for executing a process in order to synchronize said process with at least one other concurrently running process, comprising:

accessing an element of a first array of elements, said element of said first array being associated with said process, said element of said first array having an initial state;

updating said element of said first array of elements upon completion of a phase by said

process;

after said updating, accessing an element of a second array of elements, said element of said second array being associated with said process, said element of said second array having an initial hold state and being configured to switch, upon receiving an instruction, to a release state, and checking said element of said second array for said switch to said release state until detecting said release state.

26. A processor for executing a process in order to synchronize at least two concurrently running processes, said processor being operable to:

access a first array of elements, each element of said first array of elements being associated with one of said at least two concurrently running process and having an initial state;

monitor all elements of said first array of elements until detecting that each of said elements of said first array has been updated by its associated process; and

thereafter generate an instruction to switch all elements of a second array of elements from an initial hold state to a release state, each element of said second array of elements being associated with one of said at least two concurrently running processes.

27. The processor recited in claim 26, wherein said process executed thereon is one of said concurrent processes.

28. A computer program product for synchronizing at least two concurrently running processes in a data processing system, the computer program product comprising:

a computer useable medium having computer readable program code means embodied in the medium for synchronizing at least two concurrently running processes, the computer program code means including:

computer readable program code means for providing a first array of elements with initialized states, each element of said first array having a concurrently running process associated therewith, each element of said first array being configured to have its state updated by its associated concurrently running process upon completion of a phase by said associated concurrently running process;

computer readable program code means for providing a second array of elements with initialized hold states, each element of said second array having a concurrently running process associated therewith, each element of said second array being configured to switch, upon receiving an instruction, to a release state;

computer readable program code means for monitoring said first array of elements and, upon each element of said first array having had its state updated, generating said instruction for switching said elements of said second array to said release state.

29. The computer program product recited in claim 28, further comprising:

computer readable program code means for configuring each process of said at least two concurrently running processes, upon completion of said phase and upon updating of its associated element of said first array, to wait at its associated element of said second array for said release state.

30. A system for synchronizing at least two concurrently running processes in a data processing system, comprising:

(a) means for providing a first array of elements with initialized states, each element of said first array having a concurrently running process associated therewith, each element of said first array being configured to have its state updated by its associated concurrently running process upon completion of a phase by said associated concurrently running process;

(b) means for providing a second array of elements with initialized hold states, each element of said second array having a concurrently running process associated therewith, each element of said second array being configured to switch, upon receiving an instruction, to a release state;

(c) means for monitoring said first array of elements and, upon each element of said first array having had its state updated, generating said instruction for switching said elements of said second array to said release state.

31. The system recited in claim 30, further comprising:

(d) means for each process of said at least two concurrently running processes, configuring said each process such that, upon completion of said phase and upon updating of its associated element of said first array, said each process then waits at its associated element of said second array for said release state.